Лабораторная работа № 4

Использование технологии ADO.Net для доступа к данным

Цель работы: изучить различные возможности технологии ADO.Net для работы с базами данных

**Подключение к базе данных. Объект SQLConnection**

**Задание 1.** Для выполнения заданий будем использовать учебную базу данных sample.

Создайте строку подключения к базе данных, подключитесь к БД и выведите информацию о подключении. Используйте конструкцию using при создании подключения и try catch при открытии БД.

Пример:

using System.Data.SqlClient;

string connectionString = @"Data Source=DESKTOP-B2PN12T\SQLEXPRESS;Initial Catalog=sample;Integrated Security=True;";

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

connection.Open();

Console.WriteLine("Open");

// Вывод информации о подключении

Console.WriteLine("Свойства подключения:");

Console.WriteLine("\tСтрока подключения: {0}", connection.ConnectionString);

Console.WriteLine("\tБаза данных: {0}", connection.Database);

Console.WriteLine("\tСервер: {0}", connection.DataSource);

Console.WriteLine("\tВерсия сервера: {0}", connection.ServerVersion);

Console.WriteLine("\tСостояние: {0}", connection.State);

Console.WriteLine("\tWorkstationld: {0}", connection.WorkstationId);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

Проверьте, какие сообщения будут выводится, если строка подключения будет задана неверно.

**Задание 2**. Создайте какое-либо имя входа для базы данных. Подключитесь к базе данных используя это имя входа.

**Задание 3.** Создайте подключение к базе данных, используя фай конфигурации App.config. Добавьте в этот файл раздел <connectionStrings>, где задайте две строки подключения, одну для аутентификации под именем пользователя Windows, а другую под созданным именем пользователя.

Пример:

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

    <startup>

        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6" />

    </startup>

  <connectionStrings>

   <add name="DefaultConnection" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=usersdb;Integrated Security=True"

       providerName="System.Data.SqlClient"/>

  </connectionStrings>

</configuration>

Теперь получим эту строку подключения в приложении:

using System;

using System.Configuration;

namespace AdoNetConsoleApp

{

    class Program

    {

        static void Main(string[] args)

        {

            //string connectionString = @"Data Source=.\SQLEXPRESS;Initial Catalog=usersdb;Integrated Security=True";

            // получаем строку подключения

            string connectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

            Console.WriteLine(connectionString);

            Console.Read();

        }

    }

}

**Задание 4**. Создайте строку подключения с помощью конструктора SqlConnectionStringBuilder.

Пример:

SqlConnectionStringBuilder bldr = new SqlConnectionStringBuilder();

bldr.DataSource = @"SQLEXPRESS";

bldr.InitialCatalog = "userdb";

bldr.IntegratedSecurity = true;

using (SqlConnection cn = new SqlConnection(bldr.ConnectionString))

**Разработка запросов к БД. Объект SQLCommand**

**Задание 5.** Выполнение запроса, возвращающего строки.

Пример:

string connectionString= @"Data Source=LAPTOP-MNMOBLV1\SQLEXPRESS;Initial Catalog=sample;Integrated Security=True;";

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

connection.Open();

string sqlExpress = "Select \* from employee";

SqlCommand cmd = new SqlCommand(sqlExpress,connection);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.HasRows)

{

Console.WriteLine($"{reader.GetName(0)}\t{reader.GetName(1)}\t{reader.GetName(2)}\t{reader.GetName(1)}");

while (reader.Read())

{

Console.WriteLine($"{reader.GetValue(0)}\t{reader.GetValue(1)}\t{reader.GetValue(2)}\t{reader.GetValue(1)}");

}

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

Используя базу данных sample, разработайте приложение, выполняющее следующие запросы:

1. Выполните выборку из всех таблиц базы данных
2. Выполните выборку Фамилии, имени сотрудника, название проекта, дату начала работы над проектом, название должности всех сотрудников проекта p1 с ведущими должностями (т. е. аналитик — analyst и менеджер — manager).
3. Выполните выборку Фамилии, имени сотрудника, чьи отделы расположены в Сиэтле (Seattle).
4. Выполните выборку имен и фамилий всех сотрудников, работающих в отделе Research или Accounting.
5. Выполните выборку всех дат начала работы для всех клерков (clerk), работающих в отделе d1.
6. Выполните выборку всех проектов (название проекта, бюджет проекта, количество сотрудников), над которыми работают двое или больше сотрудников с должностью клерк (clerk).

**\*Разработайте метод, который позволяет выводить данные по запросу. Параметры метода- строка запроса, строка подключения.**

**Задание 6.** Получение единственного значения.

Если в запросе предусмотрено получение единственного значения, то можно воспользоваться методом ExecuteScalar, который возвращает не SqlDataReader, а первое значение первой строки обобщенного типа Object.

Выполните следующие запросы:

1. Вычисление среднего значения бюджета для всех бюджетов, превышающих $100000
2. Вычислите общую сумму бюджетов всех проектов

**Задание 7**. Выполнение запроса, не возвращающего записей. Класс SqlCommad предоставляет метод ExecuteNonQuery , который выполняет запрос, не возвращая при этом объект SqlDataReader. Эти запросы используются, когда необходимо выполнить команды Insert, Update, Delete.

Выполните следующие запросы

1. Вставьте данные для новой сотрудницы по имени Julia Long, табельный номер 11111. Она еще не назначена в какой-либо отдел.
2. Измените должности всех менеджеров (Manager) в проекте p1 на клерков (Clerk).
3. Измените значение бюджетов всех проектов на значение NULL.
4. Измените должность сотрудника с табельным номером 28559 на менеджера (Manager) для всех его проектов.
5. Повысьте на 10% бюджет проекта, менеджер которого имеет табельный номер 10102.
6. Удалите все отделы, расположенные в Сиэтле (Seattle).